PATENT APPLICATION
Docket No.: 3003.2.9A

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:	Sanchaita Datta and Ragula Bhaskar
Serial No.:	10/034,197
Filed:	December 28, 2001
For:	Combining Connections for Parallel Access to Multiple Frame Relay and Other Private Networks
Art Unit:	2453
Examiner:	Thu Ha T Nguyen

APPELLANT'S BRIEF

Honorable Commissioner for Patents:

In response to a Final Office Action mailed July 7, 2010 and pursuant to a Notice of Appeal and Extension Petition filed December 6, 2010, and 37 C.F.R. §§ 41.30 *et seq.*, Assignee appeals to the Board for relief from decisions of the Examiner.

Real Party in Interest

The real party in interest in this appeal is Assignee FatPipe Networks.

Related Appeals and Interferences

There are no pending related appeals or interferences. The Board rendered a decision July 8, 2008 (Appeal 2008-0069) regarding different claims of this application.

Status of Claims

Claims 22-40 are pending, are rejected, and are appealed.

Status of Amendments

No claim amendment was filed after final rejection.

Summary of Claimed Subject Matter

The claimed invention relates to computer network data transmission, and more particularly relates to tools and techniques for point-to-point or switched connection communications such as those using two or more frame relay networks in parallel to provide benefits such as load balancing across network connections, greater reliability, and increased security by concurrently sending different packets of the message over different network interfaces. (Application at page 1 lines 11-15, page 15 lines 8-12)

In particular, some embodiments provide the following:

22. (Figures 5-7; page 9 line 21 – page 17 line 5) A controller which controls access to multiple independent networks in a parallel network configuration, the controller comprising:

a site interface (702) connecting the controller to a site by a single logical connection;

at least two network interfaces (706) connecting the controller to respective independent parallel networks; and

a packet path selector (704) which selects between the network interfaces to split a message from the site between the networks by concurrently sending different packets of the message over different network interfaces without requiring firewall usage;

whereby the controller uses multiple networks to concurrently carry different pieces of a given message so that unauthorized interception of message packets on fewer than all of the networks used to carry the message will not provide the total content of the message.

33. (Figure 8; page 17 line 6 – page 20 line 17) A method for combining connections for access to multiple parallel networks, the method comprising the steps of:

a controller receiving (804) packets of a message sent from a site over a single logical connection, the controller having a site interface, at least two network interfaces, and a packet path selector; and

the controller packet path selector selecting (806) between the network interfaces to split the message between parallel networks by concurrently sending different packets of the message over different network interfaces, without requiring firewall usage.

40. (Figure 8; page 17 line 6 – page 20 line 17) A method for combining connections for access to multiple independent parallel frame relay networks, the method comprising the steps of:

sending (814) packets of a message over a single logical connection to a site interface of a controller, the controller having the site interface which receives packets, at least two network interfaces, and a packet path selector which selects between the network interfaces to split (812) the message between the networks by concurrently_sending different packets of the message over different network interfaces without requiring firewall usage; and

specifying at least one of the following criteria (Page 14 line 18 – page 15 line 23) for use by the packet path selector: a reliability criterion, a load-balancing criterion.

Note that the drawing reference numbers refer not only to the drawings but also to the specific locations in the text where the reference numbers are recited. The Office can readily determine those locations by searching a copy of the application. Also, the citations to drawings and text above are only examples; other parts of the specification may also be pertinent.

Grounds of Rejection to be Reviewed on Appeal

1. Claims 22, 33, and 40 were rejected in the Response to Arguments on the basis that "cited prior art teaches or suggests the subject matter broadly recited", with Kitai (US 5948069) being the only reference actually cited in the Response to Arguments.
2. Claims 33, 35, and 40 were rejected under 35 U.S.C. §102(e) as anticipated by Kitai.
3. Claims 22, 24-25, and 29 were rejected under 35 U.S.C. §103(a) as being unpatentable over Kitai in view of Dutta (US 6546423).
4. Claims 23, 28, and 30-32 were rejected under 35 U.S.C. §103(a) as being unpatentable over Kitai and Dutta in view of Albright (US 6209039).
5. Claims 26 and 27 were rejected under 35 U.S.C. §103(a) as being unpatentable over Kitai and Dutta in view of Goldszmidt (US 6195680).
6. Claim 34 was rejected under 35 U.S.C. §103(a) as being unpatentable over Kitai and Albright.

7. Claims 36-37 and 39 were rejected under 35 U.S.C. §103(a) as being unpatentable over Kitai in view of Pearce (US 5910951).
8. Claim 38 was rejected under 35 U.S.C. §103(a) as being unpatentable over Kitai, Pearce and Albright.

Argument

For purposes of this appeal only, the claims are grouped as follows:

Group I: Claims 22-34, 36-39

Group II: Claim 35

Group III Claim 40

Section 101

For clarity of the record, the undersigned notes that the January 21, 2010 Office Action asserted a Section 101 rejection asserting that "sending packets of message … could be completely performed mentally…." That rejection is not mentioned at all in the Final Office Action, and the undersigned therefore assumes the rejection has been withdrawn.

Ground 1 (Claims 22, 33, and 40)

The Final Office Action relies heavily on two errors. Every claim rejection (grounds 1-8, claims 22-40) relies on an erroneous view of the specification's actual teachings about concurrency and message splitting, and on an erroneous interpretation of the term "concurrently".

The first error occurs when the Final Office Action asserts on page 2 that the present application's specification does not teach or disclose "concurrently sending different packets of the message over different network interfaces…." Assignee respectfully disagrees.

For example, the specification teaches and discloses:

> Another difference between the inventive approach and prior approaches may also be noted here, namely, the narrow focus of some prior art on reliability differs from the present document's broader view, which considers load balancing and security as well as reliability. Configurations like those shown in Figure 2 are directed to reliability (which is

also referred to by terms such as "fault tolerance", "redundancy", "backup", "disaster recovery", "continuity", and "failover"). That is, one of the network paths (in this case, the one through the frame relay network) is the primary path, in that it is normally used for most or all of the traffic, while the other path (in this case, the one through the ISDN link) is used only when that primary path fails. Although the inventive configurations can be used in a similar manner, with one frame relay network being on a primary path and the other network(s) being used only as a backup when that first network fails, the **inventive configurations also permit concurrent use of two or more frame relay networks**. **With concurrent use, elements such as** load balancing between frame relay networks, and **increased security by means of splitting pieces of a given message between frame relay networks,** which are not considerations in the prior art of Figure 2, **become possibilities in some embodiments of the present invention**.

Page 10 line 18 – page 11 line 10 (emphasis added)

Concurrent transmission of different pieces of a message over different networks is also taught and disclosed elsewhere, e.g., in the discussion of Figure 8 on pages 18-19.

The Final Office Action relies heavily on the erroneous conclusion that the specification does not teach or disclose "concurrently sending different packets of the message over different network interfaces…." That erroneous conclusion is used, in combination with a mistaken interpretation of the term "concurrently", as the justification for asserting that Kitai teaches all limitations of claims 33, 35, and 40. Kitai's supposed teachings are also relied on in rejecting every other pending claim.

The second error also occurs on page 2, when the Final Office Action treats "concurrently" and "parallel" as if they mean the same thing. The claim specifically requires "**concurrently** sending different packets of the message over different network interfaces" whereas the Final Office Action states that Kitai teaches "**parallel** SEND/RECEIVE". As pointed out in the April 7, 2010 Response, "parallel" is not the same as "concurrent". The Final Office Action acknowledges on page 2 that this argument was presented, but fails to rebut it.

The Final Office Action provided no evidence that "parallel" in Kitai has the same meaning as "concurrent" in the present application. Moreover, even if some document confusing

"parallel" and "concurrent" were provided by the Examiner, Applicants are entitled to be their own lexicographers, and any doubts regarding which interpretation of a claim term is correct must be resolved in favor of Applicants' interpretation so long as that interpretation is consistent with the specification.

The specification uses "parallel" to describe an *arrangement* of networks, e.g., "those networks are in series rather than in parallel" (page 5 line 6), "placing the frame relay networks in parallel" (page 5 line 9), "putting networks in parallel" (page 5 line 11), "configuring private networks in parallel" (page 5 line 16), "a parallel network configuration" (page 5 line 22), "access to multiple parallel frame relay and/or point-to-point networks" (page 6 lines 18-19), "a second private network which is parallel to and independent of the first private network" (page 7 lines 2-3), and so on for many additional instances through the specification, up to and including instances in the claims as originally filed, such as "access to multiple independent private networks in a parallel network configuration" (claim 1), "parallel private networks" (claim 6), "access to multiple parallel private networks" (claim 13), "access to multiple independent parallel frame relay networks" (claim 19), and "sensing failure of one of the parallel frame relay networks and automatically sending traffic through at least one other parallel frame relay network" (claim 21).

By contrast, the specification uses "concurrently" to describe a *use* of networks, e.g., "inventive configurations also permit concurrent use of two or more frame relay networks" (page 11 line 6), "With concurrent use, elements such as load balancing between frame relay networks" (page 11 line 7), "networks 106 will be used concurrently" (page 18 line 14).

As noted in the exhibits to the April 7, 2010 Response (which are also provided in the Evidence Appendix), "parallel" means being everywhere equidistant and not intersecting in a *spatial arrangement*, whereas "concurrently" means *using* things at the same *time*, overlapping in duration. See also the Evidence Appendix article "Concurrent and Parallel Are Not The Same", which treats parallelism as a property of a machine and concurrency as a property of a program.

The mere fact that things are arranged in parallel does not mean that they are used concurrently. It is well-known to have two network connections in parallel but use them only one at a time, e.g., using one as a primary connection and the other as a failover when the

primary connection fails. Regardless of whether Kitai teaches parallel connections, Kitai fails to teach concurrently sending packets over different network interfaces of a controller as claimed. Kitai does not even mention the word "concurrent".

At pages 2-3, the Final Office Action relies on these two errors to assert that cited prior art teaches or suggests claims 22, 33, and 40. Claims 23-32, and 34-39 are then also rejected by virtue of their dependency on independent claims and other reasons, but all those rejections likewise rest on the same two errors discussed above.

Ground 2 (Claims 33, 35, 40)

Ground 2 rejects claims 33, 35, 40 under Section 102 as anticipated by Kitai. As noted above, however, Kitai fails to disclose "**concurrently** sending different packets of the message over different network interfaces…." Kitai fails to even mention "concurrent", and this failure evidences a first reason why Kitai fails to anticipate the claims.

A second reason why Kitai fails to anticipate the claims is that each claim requires "a **single logical connection**" between the site and the inventive controller. Thus, the claims are limited such that more than one logical connection is not required. By contrast, Kitai Figure 17, and Kitai Figure 3 which is referenced in the discussion of Figure 17 at column 14 lines 21 – 51, require multiple such connections. See also Kitai column 5 lines 40 – 57, discussing "a plurality of virtual channels present" from the server 3000. Kitai's approach requires special servers. Servers having a single outgoing connection will not operate as taught by Kitai. By contrast, special servers having multiple connections or multiple buffers (e.g., Kitai buffers 6031, 6032, 6033) are not required by the present invention. Servers having a single outgoing connection and otherwise configured appropriately will operate fine with the present claimed invention.

A third reason why Kitai fails to anticipate the claims is that each claim requires one to "split the message" between parallel networks. Careful reading of the cited discussion reveals that Kitai teaches splitting packets, not splitting messages. Kitai splits packets into segments based on segment lengths specified by an application; see, e.g., column 14 lines 36 – 41. By contrast, one finds no such packet segmentation requirement in the present application.

Each of these reasons provides an independent basis for reversing the rejections. Kitai fails to anticipate the claims.

Grounds 3-8 (Claims 23-32, 34, 36-39)

These grounds reject the claims under Section 103. However, the additional references cited fail to address the errors noted above. The Section 103 rejections still rely on Kitai to teach (a) concurrently sending different packets of the message over different network interfaces as claimed, (b) splitting messages as claimed, and (c) using a single logical connection as claimed. Accordingly, the Section 103 rejections should also be reversed.

Group I (claims 22-34, 36-39)

These claims were rejected under Section 102 and/or Section 103. However, all of the rejections rely on Kitai to teach (a) concurrently sending different packets of the message over different network interfaces as claimed, (b) splitting messages as claimed, and (c) using a single logical connection as claimed. As noted in the discussion of the various grounds above, Kitai fails to provide these teachings, so these claims should be allowed.

Group II (claim 35)

Claim 35 was rejected solely under Section 102 in view of Kitai. However, Kitai fails to teach (a) concurrently sending different packets of the message over different network interfaces as claimed in parent claim 33, (b) splitting messages as claimed in parent claim 33, and (c) using a single logical connection as claimed in parent claim 33. Accordingly, claim 35 should be allowed.

Group III (claim 40)

Claim 40 was rejected solely under Section 102 in view of Kitai. However, Kitai fails to teach (a) concurrently sending different packets of the message over different network interfaces as claimed, (b) splitting messages as claimed, and (c) using a single logical connection as claimed. Accordingly, claim 40 should be allowed.

Conclusion

For at least the reasons explained above, the rejections should all be reversed.

Dated January 7, 2011.

\pmAppealBrief2011-3003-2-9A

CERTIFICATE OF TRANSMISSION

I hereby certify that this Appeal Brief is being submitted to the Commissioner for Patents through EFS-WEB, on January 7, 2011.

/John Ogilvie/

Respectfully submitted,

/John W. Ogilvie/

JOHN W. OGILVIE
Registration No. 37,987

OGILVIE LAW FIRM
2552 Wilshire Circle
Salt Lake City, Utah 84109
801-706-2546 (voice)
801-583-0393 (fax)

Claims Appendix

1-21. (canceled)

22. A controller which controls access to multiple independent networks in a parallel network configuration, the controller comprising:

a site interface connecting the controller to a site by a single logical connection;

at least two network interfaces connecting the controller to respective independent parallel networks; and

a packet path selector which selects between the network interfaces to split a message from the site between the networks by concurrently sending different packets of the message over different network interfaces without requiring firewall usage;

whereby the controller uses multiple networks to concurrently carry different pieces of a given message so that unauthorized interception of message packets on fewer than all of the networks used to carry the message will not provide the total content of the message.

23. The controller of claim 22, wherein the controller controls access to multiple independent frame relay networks, and each of the at least two network interfaces comprises a frame relay network interface.

24. The controller of claim 22, wherein the packet path selector also selects between network interfaces according to a load-balancing criterion, thereby promoting balanced loads on devices that carry packets after the packets leave the selected network interfaces.

25. The controller of claim 22, wherein the packet path selector also selects between network interfaces according to a reliability criterion, thereby promoting use of devices that will still carry packets after the packets leave the selected network interfaces, when other devices that could have been selected are not functioning.

26. The controller of claim 22, wherein the controller sends packets out of sequence over the parallel networks.

27. The controller of claim 26, wherein the controller places an encrypted sequence number in at least some of the packets which are sent out of sequence.

28. The controller of claim 22, wherein the controller comprises at least three frame relay network interfaces, each of which is selectable by the packet path selector.

29. The controller of claim 22, wherein the controller operates in a system that utilizes at least one point-to-point connection.

30. The controller of claim 22, wherein the controller operates in a system providing connectivity over at least two frame relay networks from at least two carriers, each frame relay network operating on its own clock which is different from the clock of the other frame relay network.

31. The controller of claim 22, wherein each network interface is an indirect interface tailored to a particular type of frame relay network.

32. The controller of claim 22, wherein each network interface is a direct interface comprising an Ethernet card.

33. A method for combining connections for access to multiple parallel networks, the method comprising the steps of:

a controller receiving packets of a message sent from a site over a single logical connection, the controller having a site interface, at least two network interfaces, and a packet path selector; and

the controller packet path selector selecting between the network interfaces to split the message between parallel networks by concurrently sending different packets of the message over different network interfaces, without requiring firewall usage.

34. The method of claim 33, wherein the packet path selector selects between the network interfaces to split the message between parallel frame relay networks.

35. The method of claim 33, further comprising the step of specifying a load-balancing criterion for use by the packet path selector.

36. The method of claim 33, further comprising the step of specifying a reliability criterion for use by the packet path selector.

37. The method of claim 33, further comprising the steps of:

connecting the controller site interface to a site to receive packets of the message from a computer at the site over the single logical connection;

connecting a first network interface of the controller to a first network; and

connecting a second network interface of the controller to a second network which is parallel to and independent of the first network.

38. The method of claim 37, wherein at least one of the steps connecting a network interface of the controller connects the controller to a User-to-Network Interface in a router of a frame relay network.

39. The method of claim 33, further comprising the controller sensing failure of one of the parallel networks and automatically sending packets through at least one other parallel network.

40. A method for combining connections for access to multiple independent parallel frame relay networks, the method comprising the steps of:

sending packets of a message over a single logical connection to a site interface of a controller, the controller having the site interface which receives packets, at least two network interfaces, and a packet path selector which selects between the network interfaces to split the message between the networks by concurrently sending different packets of the message over different network interfaces without requiring firewall usage; and

specifying at least one of the following criteria for use by the packet path selector: a reliability criterion, a load-balancing criterion.

Evidence Appendix

(cited at Brief page 6)

WordNet Search - 3.0 - WordNet home page - Glossary - Help

Word to search for: concurrently Search WordNet

Display Options: (Select option to change) Change

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Adverb

- S: (adv) **concurrently**, at the same time (overlapping in duration) *"concurrently with the conference an exhibition of things associated with Rutherford was held"; "going to school and holding a job at the same time"*

WordNet home page

WordNet Search - 3.0 - WordNet home page - Glossary - Help

Word to search for: parallel Search WordNet

Display Options: (Select option to change) Change

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Noun

- S: (n) analogue, analog, **parallel** (something having the property of being analogous to something else)
- S: (n) latitude, line of latitude, parallel of latitude, **parallel** (an imaginary line around the Earth parallel to the equator)
- S: (n) **parallel** ((mathematics) one of a set of parallel geometric figures (parallel lines or planes)) *"parallels never meet"*

Verb

- S: (v) **parallel** (be parallel to) *"Their roles are paralleled by ours"*
- S: (v) **parallel**, collimate (make or place parallel to something) *"They paralleled the ditch to the highway"*
- S: (v) twin, duplicate, **parallel** (duplicate or match) *"The polished surface twinned his face and chest in reverse"*

Adjective

- S: (adj) **parallel** (being everywhere equidistant and not intersecting) *"parallel lines never converge"; "concentric circles are parallel"; "dancers in two parallel rows"*
- S: (adj) **parallel** (of or relating to the simultaneous performance of multiple operations) *"parallel processing"*

WordNet home page

Concurrent and Parallel Are Not The Same

Pick one: portability or efficiency. Neither is guaranteed when writing explicit parallel code

Douglas Eadline, Ph.D.
Tuesday, July 7th, 2009

In case you did not have a chance to read the column from last week, I am taking my yearly vacation at the Jersey Shore. Please refrain from the jokes, lest I pull out the Bruce Springsteen trump card. I try to spend two continuous weeks with family and friends each year. I have found that one week is just too short. I need two weeks. The first week is used to try and forget about all the stuff I did not get done before I threw the laptop in the car and say "let's go." The second week is used to try and remember and organize all the stuff I have to do when I get back. My plan usually breaks down somewhere around 10 AM my first day back to work.

This year I had a bunch of writing to do (including this column), so it was kind of a working vacation. Not to worry, I'll have my feet in the Atlantic Ocean in few hours. In any case, my dilemma is as follows. Write an insightful column quickly and get to the beach. It may surprise some readers, but I do like to research some of the topics I write about. At a minimum, I like to include enough URLs so that if you actually want to investigate a topic further, more information is just a click away. As an aside, I am constantly amazed at how much content on the web has absolutely no external links to supporting material. I thought that was the whole idea. I mean how hard is it to add a Wikipedia link to a discussion of Clos Networks or some other networking technology.

Back to my dilemma. What can talk about that will get me to beach before the water ice guy packs up for day? Although, I don't to like to rehash things I have written about in the past, I will be making a an exception this week. Not necessarily because it is easy, but because I think some messages need reinforcing. Therefore, all I have to decide is what message I should I hammer home on this July morning.

The answer is simple — understanding the difference between *concurrent* and *parallel*. I believe these two terms are often used interchangeably while, in my opinion, they are represent two different concepts.

Let's start with concurrency. A concurrent program or algorithm is one where operations can occur at the same time. For instance, a simple integration, where numbers are summed over an interval. The interval can be broken into many concurrent sums of smaller sub-intervals. As I like to say, concurrency is a property of the program. Parallel execution is when the concurrent parts are *executed* at the same time on separate processors. The distinction is

subtle, but important. And, parallel execution is a property of the machine, not the program.

If execution efficiency is important (i.e. you want things to go faster by adding more cores), then the question you need to ask is "If I run everything that is concurrent in parallel, will my code run faster?" If the answer were "yes" then we would not be having this discussion. And, since the answer, is "no", then the question is "What should run in parallel?" which is obviously, the portions of code that lower execution time.

This decision is one of the reasons cluster parallel computing is *hard*. It really does depend on the machine. Take our integration case. If the integration interval is small, then breaking it up into small sub-intervals and sending them out to other nodes will result in extending the execution time of the program due to parallel overhead. If the integration interval is huge, then parallel execution may make sense. Because parallel overhead can vary from cluster to cluster, there is no easy way to predict overhead beforehand. (i.e. The parallel overhead is larger for GigE vs InfiniBand when sending small packets.)

The same applies to multi-core. The overhead for thread communication is lower, but there is still overhead (see my HPC Hopscotch for background on SMP memory). There is no free lunch — everyone has to deal with overhead.

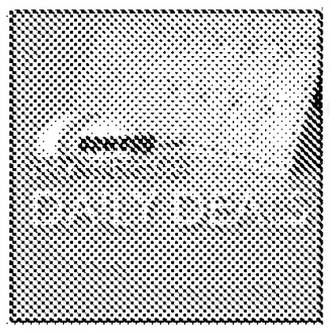
In summary, the point I want to make is this, *Concurrency is a property of the program and parallel execution is a property of the machine*. What concurrent parts *should* and *should not* be executed in parallel can only be answered when the exact hardware is known. Which I might like to add leads to the most unhappy conclusion when dealing with explicit parallel programming, *There is no guarantee of both efficiency and portability with explicit parallel programs*. Yes, I know, a sad state of affairs. I'll let you wrestle with that for a while, in the mean time, I'm going to the beach.

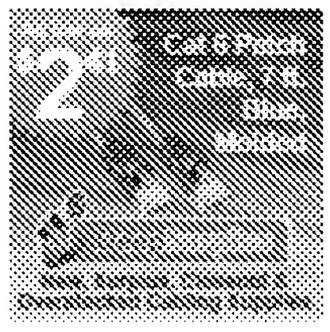
Douglas Eadline is the Senior HPC Editor for Linux Magazine

READ MORE

1. HPC 2010: The Good, The Bad, And The Ugly
2. SC10 Nuggets
3. Digging Into The Top500
4. The High Road to HPC Tools: Lunch with Intel's James Reinders
5. SMPscale: The Next HPC Challenge

COMMENTS





Related Proceedings Appendix
(none)